

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **01243126 A**

(43) Date of publication of application: **27.09.89**

(51) Int. Cl.

G06F 9/44
G06F 9/44

(21) Application number: **63071525**

(22) Date of filing: **24.03.88**

(71) Applicant: **FUJI ELECTRIC CO LTD FUJI FACOM CORP**

(72) Inventor: **ISHIBASHI KEIJI**

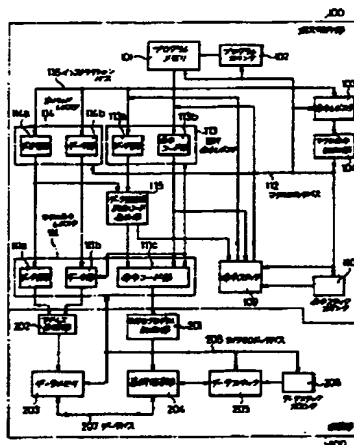
(54) ARITHMETIC AND CONTROL UNIT

(57) Abstract

PURPOSE: To execute a programming without being conscious of a data type and to dispense with the instruction of a kind according to the data type by generating a control code to indicate the discrimination and conversion methods of the data type even in an operation mutually between different data types, and executing the operation.

CONSTITUTION: A data type conversion control code generating part 115 inputs the data type code of the instruction sent from a data type part 113a of an instruction register 113 with type and the data type code of an operand sent from a data type part 114a of an operand register 114, and the part 115 generates and outputs a data conversion control code to an instruction code part 111c and the data type code of an arithmetic result to an instruction stack 109. Namely, despite the data types are different, the data types are judged, a prescribed control code and the code to indicate the data type of the arithmetic result are generated according to a conversion rule, and an arithmetic processing in the specific data type is executed. Thus, without being conscious of the data type, the programming can be executed, and an arithmetic processing speed can be improved.

COPYRIGHT: (C)1989,JPO&Japio



98 P 22 16

B7

L28 ANSWER 5 OF 9 JAPIO COPYRIGHT 1999 JPO and Japio

AN 89-243126 JAPIO

TI ARITHMETIC AND CONTROL UNIT

IN ISHIBASHI KEIJI

PA FUJI ELECTRIC CO LTD, JP (CO 000523)

FUJI FACOM CORP, JP (CO 470926)

PI JP 01243126 A 19890927 Heisei

AJ JP 88-71525 (JP63071525 Heisei) 19880324

SO PATENT ABSTRACTS OF JAPAN, Unexamined Applications, Section: P, Sect. No. 979, Vol. 13, No. 576, P. 161 (19891220)

IC ICM (4) G06F009-44

ICS (4) G06F009-44

CC 45.1 INFORMATION PROCESSING - Arithmetic sequence units

AB PURPOSE: To execute a programming without being conscious of a data type and to dispense with the ***instruction*** of a kind according to the data type by generating a control code to indicate the discrimination and ***conversion*** methods of the ***data*** ***type*** even in an operation mutually between different data types, and executing the operation.

CONSTITUTION: A ***data*** ***type*** ***conversion*** control code generating part 115 inputs the data type code of the ***instruction*** sent from a data type part 113a of an ***instruction*** register 113 with type and the data type code of an operand sent from a data type part 114a of an operand register 114, and the part 115 generates and outputs a data conversion control code to an ***instruction*** code part 111c and the data type code of an arithmetic result to an ***instruction*** stack 109. Namely, despite the data types are different, the data types are judged, a prescribed control code and the code to indicate the data type of the arithmetic result are generated according to a conversion rule, and an arithmetic processing in the specific data type is executed. Thus, without being conscious of the data type, the programming can be executed, and an arithmetic processing speed can be improved.

⑫ 公 開 特 許 公 報 (A) 平 1 - 2 4 3 1 2 6

⑤ Int. Cl. 4

識別記号

庁内整理番号

⑬ 公 開 平 成 1 年 (1 9 8 9) 9 月 2 7 日

G 0 6 F 9 / 4 4

3 2 0
3 3 0

C - 8 7 2 4 - 5 B
A - 8 7 2 4 - 5 B

審査請求 未請求 請求項の数 1 (全 1 2 頁)

⑭ 発 明 の 名 称 演 算 制 御 装 置

⑯ 特 願 昭 6 3 - 7 1 5 2 5

⑰ 出 願 昭 6 3 (1 9 8 8) 3 月 2 4 日

⑱ 発 明 者 石 橋 景 二 東 京 都 日 野 市 富 士 町 1 番 地 富 士 フ ア コ ム 制 御 株 式 会 社 内
⑲ 出 願 人 富 士 電 機 株 式 会 社 神 奈 川 県 川 崎 市 川 崎 区 田 辺 新 田 1 番 1 号
⑲ 出 願 人 富 士 フ ア コ ム 制 御 株 式 会 社 東 京 都 日 野 市 富 士 町 1 番 地
⑳ 代 理 人 弁 理 士 森 田 雄 一

明 細 書

1. 発 明 の 名 称

演 算 制 御 装 置

2. 特 許 請 求 の 範 囲

関数型で記述されたプログラムの中間言語の構文を解析し、演算部により実行可能な命令を生成する構文解析部を備え、前記中間言語を直接実行可能とした演算制御装置において、

前記構文解析部に、前記プログラムの演算命令を構成するオペランドのデータ型を判断してこのオペランドが処理されるべきデータ型への変換方法を示す制御コードと、前記オペランドを用いた演算結果のデータ型を示すデータ型コードとを発生する手段と、

演算順序を制御し、かつ前記データ型コード及び演算命令の命令コードからなるデータ型付命令をスタックする記憶手段とを備え、

前記データ型付命令を次の演算の命令として実行することを特徴とする演算制御装置。

3. 発 明 の 詳 細 な 説 明

(産 業 上 の 利 用 分 野)

本発明は、関数型で記述されたプログラムの中間言語を直接実行する演算制御装置に関する。

(従 来 の 技 術)

従来この種の演算制御装置は、例えば第4図に示すように構成されている。すなわち同図において、演算制御装置は、プログラムの中間言語を演算部200'が実行可能な形式に展開する構文解析部100'と、その展開された実行形式の命令(マクロ命令)により演算を行う演算部200'とから構成されている。

このうち構文解析部100'は、機械語化された中間言語のプログラムが格納されるプログラムメモリ101と、そのプログラムのアドレスを指定するプログラムカウンタ102と、プログラムメモリ101から読み出された命令が格納される命令レジスタ103と、演算部200'によって直接実行可能な形式の命令に展開するための制御を行うマクロ命令制御部104と、命令の実行順序を制御するための命

命令スタック109'と、そのポインタとしての命令スタックポインタ110と、実行形式の命令が格納されるマクロ命令レジスタ111'と、マクロロジックパス112とからなっている。なお、マクロ命令レジスタ111'は、データが格納されるデータ部111bと命令コードが格納される命令コード部111cとから構成されている。

一方、演算部200'は、前記命令コード部111cに接続されたマイクロプログラム制御部201と、前記データ部111bに接続されて論理アドレスから物理アドレスへの変換を行うアドレス変換部202と、データメモリ203と、算術演算部(A L U)204と、データスタック205と、そのポインタとしてのデータスタックポインタ206と、算術演算部204及びデータメモリ203間でデータの授受を行うデータバス207と、マイクロロジックバス208とからなっている。

ここで、命令スタック109'及び命令スタックポインタ110、並びにデータスタック205及びデータスタックポインタ206は、何れもL I F O (last -

in - first - out) スタックとして構成している。

次に、この動作を説明する。まず、構文解析部100'の動作を図5図のフローチャートに沿って説明すると、始めに、マクロ命令制御部104により、マクロロジックパス112を通してプログラムカウンタ102にプログラムアドレスの初期値をセットし(ステップS 1)、このアドレスに従ってプログラムメモリ101から取り出した一語の命令を命令レジスタ103に格納する(同S 2)。

次いで、命令レジスタ103の内容を解析し(同S 3)、これが命令の場合にはマクロ命令レジスタ111'の命令コード部111c及び命令スタック109'に格納し(同S 41)、プログラムカウンタ102をインクリメント(+1)して(同S 5)次の一語の構文解析に移る。また、命令レジスタ103の内容がオペランドの場合には、命令レジスタ103の内容をデータ部111bに格納し(同S 42)、命令スタック109'の先頭の命令を読み出して命令コード部111cに格納した後(同S 43)、演算部200'に対して演算処理の起動をかけ(同S 44)、その後、プログラムカウ

- 3 -

ンタ102をインクリメントする(同S 5)。

なお、次の命令の処理は、演算部200'の演算処理が終了し次第、プログラムメモリ101から命令を取り出し、上記の処理が命令順に繰返し実行される。

そして演算部200'においては、まず、マイクロプログラム制御部201がマクロ命令レジスタ111'の命令コード部111cから命令コードを読み出してデコードすると共に、この命令が、データ部111bの内容をアドレスとしたデータメモリ203からの読み出し命令であれば、アドレス変換部202を介してデータメモリ203からデータを読み出し、かかるデータをデータバス207及び算術演算部204を介してデータスタック205に書き込む。

また、前記命令が、データ部111bの内容をアドレスとした書き込み命令であるならば、上記とは逆にデータスタック205から算術演算部204及びデータバス207を介してアドレス変換部202の示すデータメモリ203のアドレスへデータを書き込む。

そして、算術演算部204は、データスタック205

内のデータ同士の算術論理演算をも行う。

次に、具体的な演算例に従い、マクロ命令制御部104及び算術演算部204の動きを、命令・データそれぞれのL I F O スタックの動作を中心として説明する。

いま、関数型の演算例として次の(1)式を考え、この(1)式をポーランド記法を用いて(2)式のように展開する。

$$f = A + B \times (C + D) \quad \dots \dots \dots (1)$$

$$\left(\frac{\sqrt{A}}{\textcircled{1}} \frac{\frac{\times}{\textcircled{2}} \frac{B}{\textcircled{3}} \left(\frac{+}{\textcircled{4}} \frac{C}{\textcircled{5}} \frac{D}{\textcircled{6}} \right) \right) \frac{+}{\textcircled{7}} \right) \frac{=}{\textcircled{8}} \quad \dots (2)$$

そして、第4図のプログラムメモリ101には、この(2)式が機械語に展開されて第6図に示す順序①～⑧で格納されているとする。なお、この順序①～⑧は構文解析部100'による処理順序でもあり、以後は各処理自体も番号①～⑧によって表すものとする。

以下、上記処理①～⑧毎に、命令スタック109'及びデータスタック205の作用を、第7図に基づいて説明する。

まず、処理①、③、⑤は命令コードの読み出し

- 5 -

-216-

- 6 -

である
ード部
にそれ
処理の
のオペ
にデー
される
定数で
ればそ
て扱
更に
み取
ータ
された
(同)と
その
る。
ク205
C + I
次の

す
Dカ
とデ
め、
って
にデ
示す
令を
(二)
f、
205
ま
統一
に変
小数
して
それ
Dを
数点
デー

であるため、これらの命令は前述した如く命令コード部111cに格納され、同時に命令スタック109にそれぞれプッシュダウンして格納される。また、処理③、④、⑤はオペランドであるため、これらのオペランドを評価した上で、前述の如く最終的にデータスタック205にプッシュダウンして格納される。ここで、「評価する」とは、オペランドが定数であればそのままの値として、アドレスであればそのアドレスの示す内容を読み取って値として扱うことをいう。

更に、処理⑥は処理⑤に続いてオペランドが読み取られたもので、オペランドを評価した上でデータスタック205の先頭のデータ(最後にスタックされたもの)と、命令スタック109の先頭の命令(四)とを用いて算術演算部204にて演算を行い、その演算結果を再びデータスタック205に格納する。なお、第7図の処理⑥におけるデータスタック205の内容 f_1 は、演算の途中経過である $f_1 = C + D$ を示している。

次いで、処理⑦、⑧、⑨は、対応する命令の処

理の終わりを示すもので、命令スタック109の先頭の命令をポップアップした後、⑦の先頭の命令演算をデータスタック205の先頭のデータとその次に格納されたデータとを用いて行い、その演算結果をデータスタック205をポップアップした後、に再び格納する。なお、第7図の処理⑧におけるデータスタック205の内容 f_1 は、演算の途中経過である $f_1 = B \times f_1$ を示し、処理⑧における f_1 は $f_1 = A + f_1$ 、つまり(1)式の解を示している。

以上の動作により、(1)式の演算処理が終了することになる。

(発明が解決しようとする課題)

ここで、(1)式におけるデータA、B、C、Dの型が異なり、これらを用いて演算処理を行う場合には、以下の点に留意する必要がある。具体的には、A、B、Dが整数値、Cが浮動小数点値というように異なるデータ型で与えられた場合には、(1)式を、データ型を変換する命令及び特定のデータ型を対象とする命令に置き換えてプログラムしなければならない。

- 7 -

すなわち、第8図(イ)に示すようにデータCとDから $C + D$ の演算を行って f_1 を求め、この f_1 とデータBとから $B \times f_1$ の演算を行って f_2 を求め、この f_2 とデータAとから $A + f_2$ の演算を行って解 f を求める演算処理を行う場合、このようにデータ型が異なると、第8図(ロ)または(ハ)に示す如くデータ型を何れか一方に合わせる変換命令を挿入する必要がある。なお、第8図(イ)~(ニ)において、データA~D及び演算結果 f_1 、 f_2 、 f はデータメモリ203またはデータスタック205に格納された状態である。

まず、第8図(ロ)はデータ型を浮動小数点値に統一するべく、データA、B、Dを浮動小数点値に変換する命令“I/F”を挿入し、その後、浮動小数点値を対象とした演算に置き換える処理を示している。この図において、“a”、“b”、“d”はそれぞれ浮動小数点値に変換後のデータA、B、Dを、“+F”、“×F”、“÷F”はそれぞれ浮動小数点演算による加算、乗算、除算を示す。また、データa、b、dはデータメモリ203またはデー

タスタック205に格納された状態である。

一方、第8図(ハ)はデータ型を整数値に統一するべく、データCを整数値に変換する命令“F/I”を挿入し、その後、整数値を対象とした演算に置き換える処理であり、この図において、“c”は整数値に変換後のデータCを、“+”、“×”、“÷”はそれぞれ整数演算による加算、乗算、除算を示す。

更に、第8図(ニ)は、何れか一方のデータ型に統一して処理するのではなく、各データがそれぞれ値とデータ型を持ち、演算の際にデータ型を評価して演算処理を行うものであり、演算結果として同様に値とデータ型を出力し、これをスタックして次の演算に移行するものである。この図において、データA、B、C、D及び演算結果 f_1 、 f_2 、 f に付された“I”、“F”はそれぞれ整数値、浮動小数点値のデータ型を示している。なお、この方式により、先の第7図の処理⑥を行った時点でのデータスタック205の内容は第9図のとおりとなり、各データについて各々のデータ型“I”または“F”がそれぞれ格納されることになる。

- 9 -

- 217 -

- 10 -

ここで、上述した第8図(ロ)・(ハ)のように、データは値だけで命令が特定のデータ型のみを対象とした演算種類を持つ場合には、以下のような問題がある。つまりこの場合には、プログラマが、データ型を意識しながらデータ型変換命令の記述と当該データ型に応じた演算命令の選択とを行ってプログラミングしなければならず、演算命令もデータ型の種類に応じた数だけ用意しなくてはならないため、プログラミングが極めて煩雑となる。また、演算制御装置においても複数のデータ型分の命令処理が必要となり、命令の構文解析部及び演算部のハードウェア、ファームウェア処理が増大してしまう。

更に、第8図(ニ)のように、データに値とデータ型を持ち、演算の際にデータ型を評価して演算処理を行う場合には、第9図に示した如く、演算に際して、データスタック205からデータ型を取り出してその変換の要否を判断した後に実際の演算を行い、また、演算結果 f_1 、 f_2 、 f も値とデータ型とを有するため、演算結果を評価する際に

データ型を認識する時間を必要としてデータの処理時間が多くかかるという場合がある。同時に、データ型を記憶するためデータスタック205のメモリ容量を多く必要としていた。

本発明は上記問題点を解決するために提案されたものであり、その目的とするところは、異なるデータ型同士の演算において、データ型を意識することなくプログラミングを行うことができ、プログラマや演算制御装置のハードウェア等の負担を軽減できると共に、演算結果にデータ型を含ませず値のみを持たせることにより、データスタックのメモリ効率を高め、かつ、演算結果の評価に要する時間を短縮して演算処理速度の大幅な向上を可能にした演算制御装置を提供することにある。(課題を解決するための手段)

上記目的を達成するため、本発明は、関数型で記述されたプログラムの中間言語を直接実行する演算制御装置において、中間言語の構文解析部に、プログラムの演算命令を構成するオペランドのデータ型を判断してこのオペランドが処理されるべ

きデータ型への変換方法を示す制御コードと、前記オペランドを用いた演算結果のデータ型を示すデータ型コードとを発生する手段と、演算順序を制御し、かつ前記データ型コード及び演算命令の命令コードからなるデータ型付命令をスタックする命令スタック等の記憶手段とを備え、前記データ型付命令を次の演算の命令として実行することを特徴とする。

(作用)

本発明によれば、オペランドのデータ型と型付命令のデータ型とに基づき、データ型の変換方法を示す制御コードと演算結果のデータ型を示すコードとが生成される。このうち、前記制御コードにより必要に応じてデータ型の変換が行われ、型付命令の命令コードに基づいて演算が実行される。

また、前記演算結果のデータ型を示すコードは命令スタックに送られ、命令コードと共に新たなデータ型付命令としてスタックされることにより、次の命令の実行時に受け継がれることとなる。

(実施例)

以下、図に沿って本発明の一実施例を説明する。まず、第1図は本発明にかかる演算制御装置の構成を示したもので、同図において、100はプログラムの中間言語を演算部200が実行可能な形式に展開する構文解析部、200はマクロ命令により演算を行う演算部200である。

ここで、演算部200は、先の第4図に示したように、マイクロプログラム制御部201と、アドレス変換部202と、データメモリ203と、算術演算部204と、データスタック205と、このデータスタック205と共にLIFOスタックを構成するデータスタックポインタ206と、算術演算部204及びデータメモリ203間でデータの授受を行うデータバス207と、マイクロロジックバス208とからなっている。

次に、構文解析部100の構成を詳述する。この構文解析部100は、前記同様に機械語化された中間言語のプログラムが格納されるプログラムメモリ101と、このプログラムのアドレスを指定するプログラムカウンタ102と、プログラムメモリ101

から読み出
103と、演算
命令に展開
部104と、
スタック10
F Oスタック
0とを備え
。しかし
トラクシ
格納され
メモリ101
格納され
ている。
ータ型を
型部113a
される命
ランドレ
示すデー
と、論理
ータ部11

生成して
なお、
と共に型
Rを構成
演算結果
3の命令
データ型付
ている。
また、
及びデー
部202に
及びデー
レスへの
次に、
を行う
図を参照
まず、
プログ
のオペ
ットを

から読み出された命令が格納される命令レジスタ103と、演算部200によって直接実行可能な形式の命令に展開するための制御を行うマクロ命令制御部104と、命令の実行順序を制御するための命令スタック109と、この命令スタック109と共にLIFOスタックを構成する命令スタックポインタ110を備えている。

しかし、プログラムメモリ101には、インストラクションバス116を介してデータ型付命令が格納される型付命令レジスタ113と、プログラムメモリ101から読み出された命令のオペランドが格納されるオペランドレジスタ114とが接続されている。このうち、型付命令レジスタ113は、データ型を示すデータ型コードが格納されるデータ型部113aと、命令の種類を示す命令コードが格納される命令コード部113bとからなり、また、オペランドレジスタ114は、オペランドのデータ型を示すデータ型コードが格納されるデータ型部114aと、論理アドレスまたは直接数値が格納されるデータ部114bとからなっており、マクロ命令制御部

104からの信号がマクロレジスタバス112を介して型付命令レジスタ113及びオペランドレジスタ114に加えられている。

更に、111は実行形式の命令が格納されるマクロ命令レジスタであり、このレジスタ111は、前記オペランドレジスタ114のデータ型部114a及びデータ部114bにそれぞれ接続されたデータ型部111a及びデータ部111bと、型付命令レジスタ113のデータ型部113aにデータ型変換制御コード発生部115を介して接続され、かつ命令コード部113bに接続される命令コード部111cとから構成されている。なお、データ部111b及び命令コード部111cはマクロ命令制御部104にも接続されている。

ここで、データ型変換制御コード発生部115は、型付命令レジスタ113のデータ型部113aから送られる命令のデータ型コードと、オペランドレジスタ114のデータ型部114aから送られるオペランドのデータ型コードとを入力とし、命令コード部111cに対するデータ型変換制御コードと、命令スタック109に対する演算結果のデータ型コードとを

- 15 -

生成して出力するようになっている。

なお、命令スタック109はプログラムメモリ101と共に型付命令レジスタ113に対しワイヤードORを構成しており、かかる命令スタック109には演算結果のデータ型コードと型付命令レジスタ113の命令種類を示す命令コードとを合成してデータ型付命令としてスタックできるように構成されている。

また、マクロ命令レジスタ111のデータ型部111a及びデータ部111bは演算部200内のアドレス変換部202に接続され、オペランドのデータ型コード及びデータに基づいて論理アドレスから物理アドレスへの変換が行われるものである。

次に、この実施例により先の(1)式の演算処理を行う場合について、その動作を第2図及び第3図を参照しつつ説明する。

まず、第2図(イ)、(ロ)はこの実施例においてプログラムメモリ101に格納されている中間言語のオペランドのフォーマット及び命令のフォーマットをそれぞれ示しており、オペランドと命令と

はフォーマットの最上位ビットの"1"または"0"により識別されている。このうち、オペランドのフォーマットは、第2図(イ)に示すようにデータの型、つまり整数値や浮動小数点値といった型を示すデータ型コードと論理アドレスまたは直接数値とで構成され、各々そのままの値でオペランドレジスタ114を経てマクロ命令レジスタ111へ送られる。

また、命令のフォーマットは、第2図(ロ)に示すようにデータ型コードを持たず(=0)、命令種類を示す命令コードのみで構成され、前述した命令スタック109とのワイヤードORにより、型付命令レジスタ113を経てマクロ命令レジスタ111の命令コード部111cに送られる。この中でデータ型部113aについては、データ型変換制御コード発生部115によりデータ型を変換する制御コードが生成され、かかる制御コードが命令コード部111cに送られる。なお、第2図(ハ)は、命令スタック109から読み出された命令のフォーマットであり、このフォーマットは演算結果のデータ型を示すデ

- 17 -

- 219 -

- 18 -

データ型コード及び命令コードを有し、また、第2図(ニ)はマクロ命令レジスタ111の命令コード部111cのフォーマットであって、データ型変換制御コード及び命令コードからなっている。

なお、第2図(イ)～(ニ)において、各フォーマットの下に付された記号は、これらのコード等が格納される場所を示している。

ここで、前記ワイヤードORの制御について説明すると、先の第6図及び第7図の処理①、②、③は命令コードの読み出しであるため、プログラムメモリ101が選択されて命令コードのみが型付命令レジスタ113の命令コード部113bに送られる。また、処理④、⑤、⑥、⑦は演算であるから命令スタック109が選択され、データ型コード及び命令コードがデータ型部113a及び命令コード部113bに送られる。上記制御は、マクロ命令制御部104が各処理ごとに命令レジスタ103に取り込んだ内容に基づき、命令可否かを判断して行う。

次に、データ型変換制御コードの生成方法について述べる。データ型の変換は、データ語長また

はデータの表し得る幅(データレンジ)の大きい方に合わせれば、演算及び演算結果のデータ型を一義的に決めることができる。例えば、データが一語長整数値、二語長整数値及び浮動小数点値の3種類である場合、その変換ルールは、次の(3)式の大小関係で大きい方のデータ型に従うものとする。

$$\text{一語長整数値} < \text{二語長整数値} < \text{浮動小数点値} \quad \text{--- (3)}$$

この点に着目し、データ型変換制御コード発生部115は、以下の表1の如く、オペランドのデータ型コード及び型付命令のデータ型コードを入力として、データ型の変換を示す制御コードと演算結果の型を示すコードとを生成する。

(以下、余白)

表 1

入 力		出 力	
オペランドのデータ型コード	型付命令のデータ型コード	データ型の変換を示す制御コード	演算結果の型を示すコード
DI	"0"	DC	D
SI	"0"	SC	S
FP	"0"	FC	F
DI	D	DC	D
SI	D	SDC	D
FP	D	FDC	F
DI	S	DSC	D
SI	S	SC	S
FP	S	FSC	F
DI	F	DFC	F
SI	F	SFC	F
FP	F	FC	F

なお、この表1において、"0"はデータ型を持つ前のプログラムメモリ101に格納された状態値を、"DI"は二語長整数を示すオペランドコードを、"SI"は一語長整数を示すオペランドコードを、"FP"は浮動小数点値を示すオペランドコードを、"D"は二語長データを示す型コードを、"S"は一語長データを示す型コードを、"F"は浮動小数点データを示す型コードを、"DC"はデー

タ型変換なしで二語長演算を行うコードを、"SC"はデータ型変換なしで一語長演算を行うコードを、"FC"はデータ型変換なしで浮動小数点演算を行うコードを、"SDC"はオペランドのデータ型を"D"の型に変換して二語長演算を行うコードを、"FDC"はデータスタック205上のデータの"D"の型を"F"の型に変換して浮動小数点演算を行うコードを、"DSC"はデータスタック205上のデータの"S"の型を"D"の型に変換して二語長演算を行うコードを、"FSC"はデータスタック205上のデータの"S"の型を"F"の型に変換して浮動小数点演算を行うコードを、"DFC"及び"SFC"はオペランドのデータ型を"F"の型に変換して浮動小数点演算を行うコードをそれぞれ示している。

次に、この実施例によって先の(1)式に示した演算処理を行う場合、特に処理⑤、⑥、⑦について動作を詳述する。

まず、処理⑥において、データC(浮動小数点値)にかかるオペランドが読み取られると、オペ

ランドレジスタ114のデータ型部114a及びマクロ命令レジスタ111のデータ型部111aからはデータ型コード"FP"が出力され、オペランドレジスタ114のデータ部114b及びマクロ命令レジスタ111のデータ部111bからは論理アドレスまたは直接数値が出力される。このとき、型付命令レジスタ113のデータ型部113a及び命令コード部113bには、処理⑥の「(+)」の命令が格納されており、この命令は第2図(ロ)に示すようにデータ型コードを持たず(="0"), 命令コードのみを有している。

このため、データ型変換制御コード発生部115には、データ型部114aからオペランドのデータ型コード"FP"と、データ型部113aから型付命令のデータ型コード"0"とが入力され、先の表1に従ってデータ型の交換方法を示す制御コード"FC"と演算結果のデータ型を示すコード"F"とが出力される。このうち、制御コード"FC"はマクロ命令レジスタ111の命令コード部111cに送られ、また、演算結果のデータ型を示すコード"F"は命令スタック109に送られて命令コードと共に次のデ

ータ型付命令を構成する。

次いで、データD(整数値)にかかる処理のによりオペランドが読み取られると、オペランドのデータ型コード"D I"がデータ型部114a、111aから出力され、論理アドレスまたは直接数値がデータ部114b、111bから出力されると共に、データ型コード"D I"がデータ型変換制御コード発生部115に入力され、かつ、命令スタック109からデータ型部113aを介して演算結果のデータ型を示すコード"F"が入力される。

従って、表1により、データ型変換制御コード発生部115からはデータ型の交換方法を示す制御コード"DFC"と演算結果の型を示すコード"F"とが出力され、これらは前記同様に命令コード部111c及び命令スタック109にそれぞれ送られる。ここで、制御コード"DFC"はオペランドを浮動小数点データに変換して浮動小数点演算を行う旨のコード、コード"F"は演算結果を浮動小数点データとして得るためのコードである。

すなわち、処理⑥、⑦の各データC、Dはデー

- 23 -

タ型が異なるにも拘らず、データ型変換制御コード発生部115でこれらのデータ型を判断し、先の(3)式の変換ルールに従って所定の制御コード及び演算結果のデータ型を示すコードを生成して特定のデータ型での演算処理が実行されることになる。

一方、演算部200では、先のデータCについてデータ型部111a及びデータ部111bの内容に基づきアドレス変換部202により物理アドレスへの変換を行い、データメモリ203から読み出してデータスタック205に格納したデータと、同じくデータDについて同様の作用により読み出したデータとを用い、命令コード部111cからの型付の命令コードをマイクロプログラム制御部201によりデコードして算術演算部204により演算を行い、その結果($f_1 = C + D$)をデータスタック205に格納する。この演算結果にはデータ型が付加されていないが、演算結果のデータ型は、上述したように命令スタック109によって管理されることになる。

なお、演算部200によって上記演算が実行され

- 24 -

ている間、構文解析部100ではオペランドレジスタ114及び型付命令レジスタ113等により次の命令の構文解析が並行して行われている。

第3図は、上述したようにオペランドのデータ型が型付の命令として受け継がれる様子を示したものであり、図中、符号 Y_1 、 Y_2 、 Y_3 はデータ型の比較判別を行ってデータ型変換制御コードを生成するデータの流れ、また、符号 i 、 ii 、 iii はデータ型が引き継がれる順序を、更に④は生成されたデータ型とデータ型付命令であることをそれぞれ示している。

この実施例において、マクロ命令レジスタ111を省略してその処理を型付命令レジスタ113及びオペランドレジスタ114に代行させることもできる。また、これらの型付命令レジスタ113及びオペランドレジスタ114は構文解析処理と演算処理とを並行して行わせるために設けられており、かかる並行処理の必要がなければ省略することも可能である。

(発明の効果)

- 25 -

-221-

- 26 -

以上詳述したように本発明によれば、プログラムを構成するオペランドにデータ型を示すコードを付加し、一方、命令コードは特定なデータ型を持たない命令語の構造とし、異なるデータ型同士の間での演算においてもそのデータ型の判別及び変換方法を示す制御コードを生成して演算を行わせるようにしたから、データ型を意識することなくプログラミングを行うことができ、また、データ型に応じた種類の命令を備える必要がないことから、プログラミングやハードウェア上の負担を軽減し、しかもプログラムメモリ容量の削減を図ることができる。

更に、演算結果のデータ型を示すコードを生成し、このコードと演算の命令コードとを合成してデータ型付命令を構成すると共に、このデータ型付命令をスタックして次の演算データの命令として再び実行できるようにしたため、データ型が次命令に受け継がれることになり、演算結果を格納するデータスタックではデータ型を除いた値だけ格納すれば済むため、スタックのメモリ容量を

少なくすることが可能である。

従って、演算結果を評価するデータ型の認識のための処理を省くことができ、演算の処理速度が向上する等の効果がある。

4. 図面の簡単な説明

第1図ないし第3図は本発明の一実施例を示すもので、第1図はブロック図、第2図は(イ)～(ニ)は各フォーマットの説明図、第3図は演算処理の一例においてデータの流れを抽象的に表現した説明図、第4図は従来例を示すブロック図、第5図は構文解析部の動作を示すフローチャート、第6図はプログラムメモリの内容を示す説明図、第7図は各処理ごとの命令スタック及びデータスタックの内容を示す説明図、第8図(イ)～(ニ)は演算処理の一例においてデータの流れを抽象的に表現した説明図、第9図はデータスタックの内容の説明図である。

100…構文解析部

113…型付命令レジスタ

109…命令スタック

111…マクロ命令レジスタ

- 27 -

111a…データ型部 111b…データ部
113a…データ型部 113b…命令コード部
114…オペランドレジスタ 114a…データ型部
114b…データ部
115…データ型変換制御コード発生部
200…演算部

特許出願人

富士電機株式会社
(外1名)

代理人

井理士

森田 雄



- 28 -

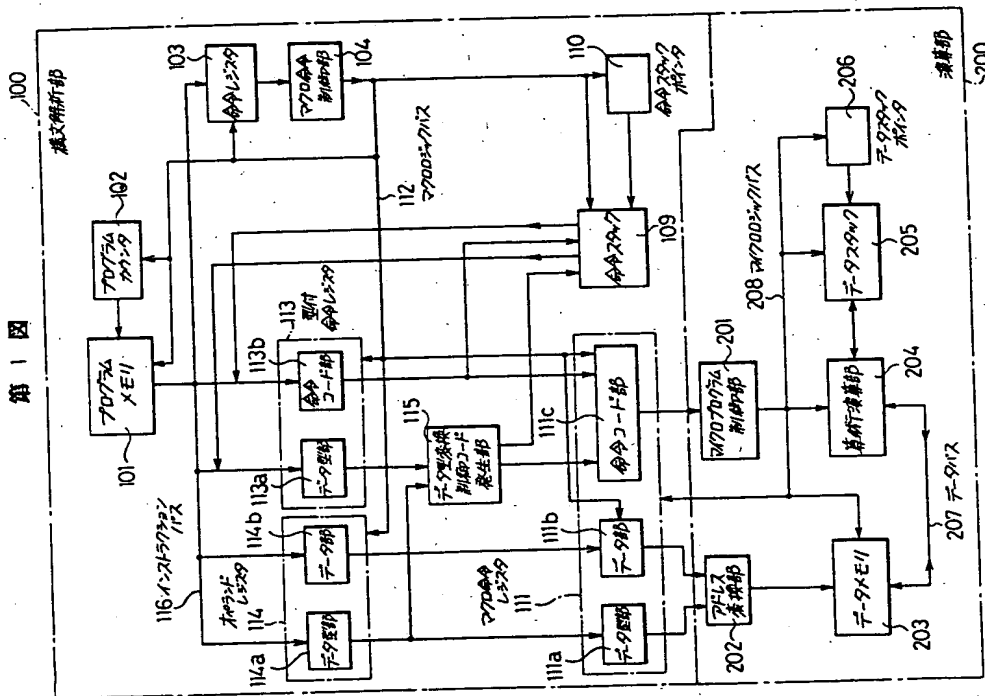
- 29 -

- 222 -

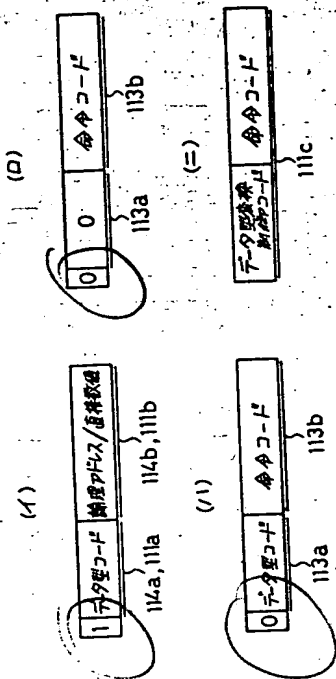
100

第1図

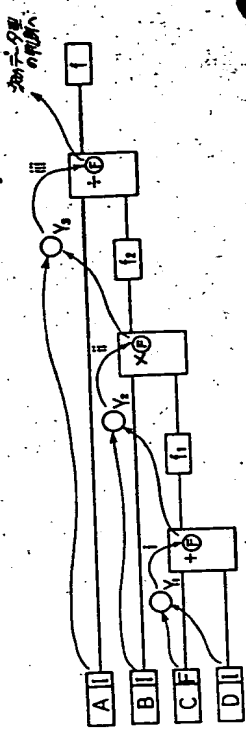
第 1 図



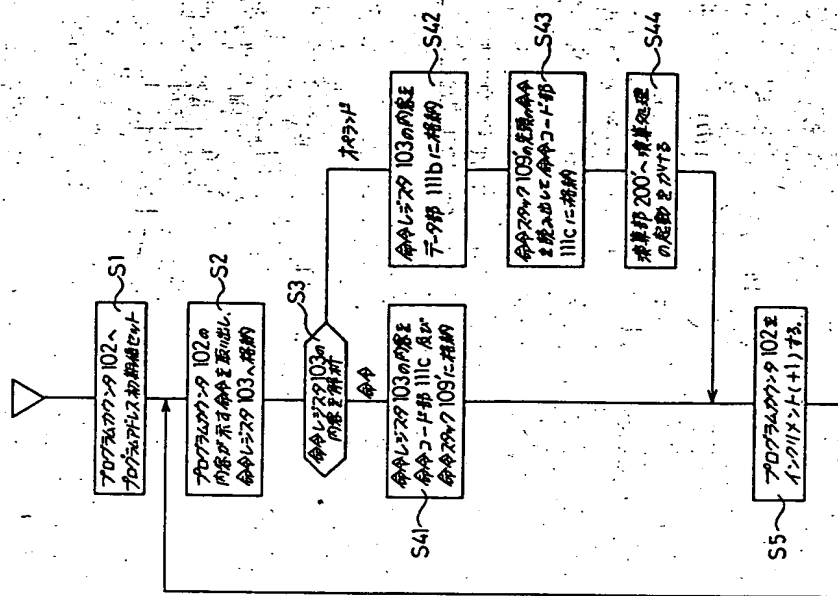
第 2 図



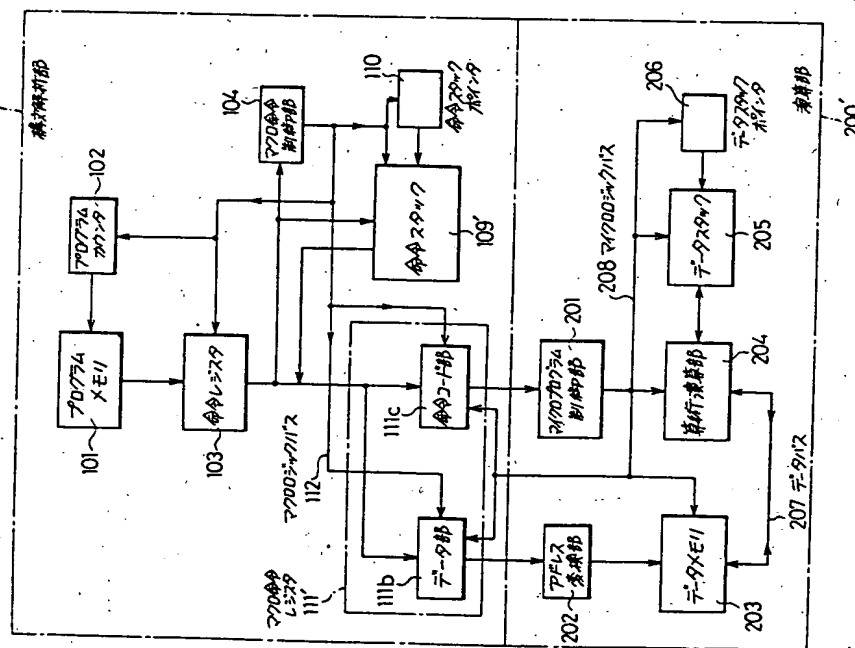
第 3 図



第 5 図

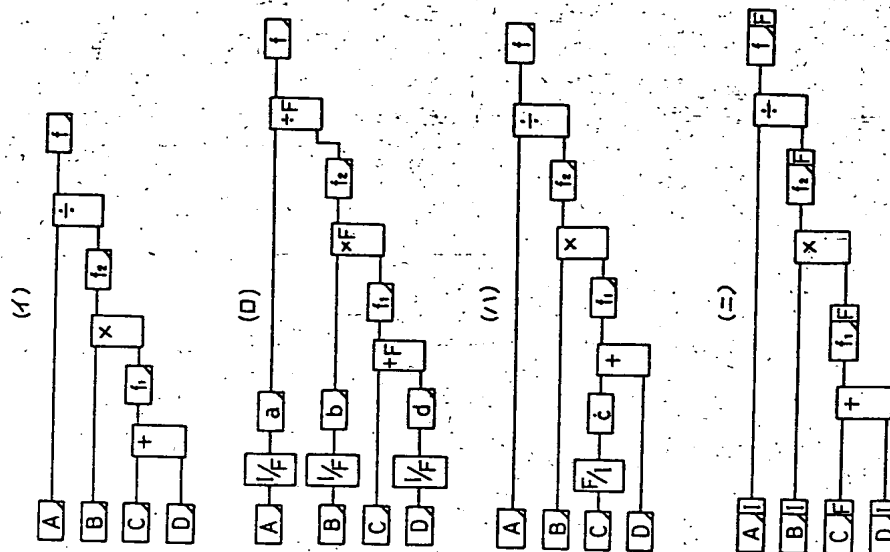


第 4 図

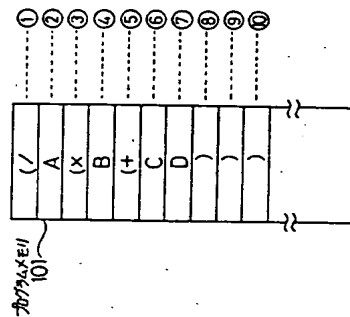


第 8 図

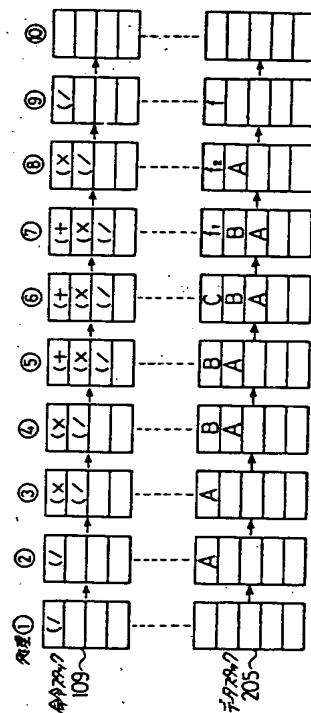
第 8 圖



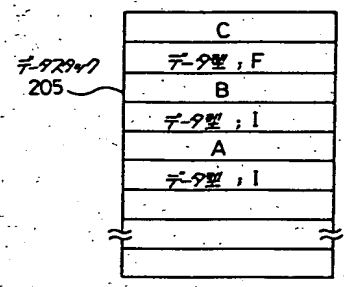
第 6 圖



第 7 圖



第 9 図



Int. Cl.
G 06 F

④発明の名

④発 明

④発 明

④出 願

④代 理

1. 発明のイ

論理型

2. 特許種別

1. 論理型

ムに変更

(1) [

れ、

おこ

グラ

第

(2) [

だし

(

1 の

数デ

(

くり

第

(3) [

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.